TMS320C674x/OMAP-L1x Processor Peripherals Overview

Reference Guide

TEXAS INSTRUMENTS

Contents

Read This First

About This Manual

This document provides an overview and briefly describes the peripherals available on the TMS320C674x Digital Signal Processors (DSPs) and OMAP-L1x C6-Integra DSP+ARM Processors. For details on a specific device or peripheral, see the device-specific technical reference manual.

Related Documentation From Texas Instruments

The following documents describe the TMS320C674x DSP and OMAP-L1x C6-Integra DSP+ARM Processor. Copies of these documents are available on the Internet at www.ti.com. *Tip:* Enter the literature number in the search box provided at www.ti.com.

The current documentation that describes the DSP, related peripherals, and other technical collateral, is available in the C6000 DSP product folder at: www.ti.com/c6000.

SPRUH77 — *OMAP-L138 C6-Integra DSP+ARM Processor Technical Reference Manual.* Describes the System-on-Chip (SoC) and each peripheral in the device. The SoC includes the ARM subsystem and associated memories, the DSP subsystem and associated memories, and a set of I/O peripherals.

SPRUH78 — *OMAP-L132 C6-Integra DSP+ARM Processor Technical Reference Manual.* Describes the System-on-Chip (SoC) and each peripheral in the device. The SoC includes the ARM subsystem and associated memories, the DSP subsystem and associated memories, and a set of I/O peripherals.

SPRUH79 — *TMS320C6748 DSP Technical Reference Manual.* Describes the System-on-Chip (SoC) and each peripheral in the device. The SoC includes the DSP subsystem and associated memories, and a set of I/O peripherals.

SPRUH80 — *TMS320C6746 DSP Technical Reference Manual.* Describes the System-on-Chip (SoC) and each peripheral in the device. The SoC includes the DSP subsystem and associated memories, and a set of I/O peripherals.

SPRUH81 — *TMS320C6742 DSP Technical Reference Manual.* Describes the System-on-Chip (SoC) and each peripheral in the device. The SoC includes the DSP subsystem and associated memories, and a set of I/O peripherals.

SPRUH90 — *TMS320C6743 DSP Technical Reference Manual.* Describes the System-on-Chip (SoC) and each peripheral in the device. The SoC includes the DSP subsystem and associated memories, and a set of I/O peripherals.

SPRUH91 — *TMS320C6745/C6747 DSP Technical Reference Manual.* Describes the System-on-Chip (SoC) and each peripheral in the device. The SoC includes the DSP subsystem and associated memories, and a set of I/O peripherals.

SPRUH92 — *OMAP-L137 C6-Integra DSP+ARM Processor Technical Reference Manual.* Describes the System-on-Chip (SoC) and each peripheral in the device. The SoC includes the ARM subsystem and associated memories, the DSP subsystem and associated memories, and a set of I/O peripherals.

SPRUFK5 — *TMS320C674x DSP Megamodule Reference Guide.* Describes the TMS320C674x digital signal processor (DSP) megamodule. Included is a discussion on the internal direct memory access (IDMA) controller, the interrupt controller, the power-down controller, memory protection, bandwidth management, and the memory and cache.

SPRUFE8 — *TMS320C674x DSP CPU and Instruction Set Reference Guide.* Describes the CPU architecture, pipeline, instruction set, and interrupts for the TMS320C674x digital signal processors (DSPs). The C674x DSP is an enhancement of the C64x+ and C67x+ DSPs with added functionality and an expanded instruction set.

SPRUG82 — *TMS320C674x DSP Cache User's Guide.* Explains the fundamentals of memory caches and describes how the two-level cache-based internal memory architecture in the TMS320C674x digital signal processor (DSP) can be efficiently used in DSP applications. Shows how to maintain coherence with external memory, how to use DMA to reduce memory latencies, and how to optimize your code to improve cache efficiency. The internal memory architecture in the C674x DSP is organized in a two-level hierarchy consisting of a dedicated program cache (L1P) and a dedicated data cache (L1D) on the first level. Accesses by the CPU to the these first level caches can complete without CPU pipeline stalls. If the data requested by the CPU is not contained in cache, it is fetched from the next lower memory level, L2 or external memory.

TMS320C674x/OMAP-L1x Processor Peripherals Overview

1 Overview

The user-accessible peripherals available on the TMS320C674x DSP and OMAP-L1x C6-Integra DSP+ARM Processor are configured using a set of memory-mapped control registers. The peripheral bus controller performs the arbitration for accesses of on-chip peripherals. Peripherals available on the TMS320C674x DSP are listed in Table 1. Peripherals available on the OMAP-L1x C6-Integra DSP+ARM Processor are listed in Table 2. For details on a specific device or peripheral, see the device-specific technical reference manual.

Table 1. TMS320C674x DSP Peripherals Documentation

Peripheral/Module	Acronym	C67x					
		42	43	45	46	47	48
DDR2/mDDR Memory Controller	DDR2	√	—	—	√	—	√
DSP Interrupt Controller	INTC	√	√	√	√	√	√
Enhanced Capture Module	eCAP	√	√	√	√	√	√
Enhanced Direct Memory Access Controller	EDMA3	√	√	√	√	√	√
Enhanced High-Resolution Pulse-Width Modulator	eHRPWM	√	√	√	√	√	√
Enhanced Quadrature Encoder Pulse Module	eQEP	—	√	√	—	√	—
Ethernet Media Access Controller/ Management Data Input/Output Module	EMAC/MDIO	—	√	√	√	√	√
External Memory Interface A	EMIFA	√	√	√	√	√	√
External Memory Interface B	EMIFB	—	√	√	—	√	—
General-Purpose Input/Output	GPIO	√	√	√	√	√	√
Host Port Interface	HPI	√	—	—	√	√	√
Inter-Integrated Circuit Module	I2C	√	√	√	√	√	√
Internal Direct Memory Access Controller	IDMA	√	√	√	√	√	√
Liquid Crystal Display Controller	LCDC	—	—	—	—	√	√
Multichannel Audio Serial Port	McASP	√	√	√	√	√	√
Multichannel Buffered Serial Port	McBSP	√	—	—	√	—	√
Multimedia Card/Secure Digital Card Controller	MMC/SD	—	√	√	√	√	√
Phase-Locked Loop Controller	PLLC	√	√	√	√	√	√
Power-Down Controller	PDC	√	√	√	√	√	√
Power and Sleep Controller	PSC	√	√	√	√	√	√
Programmable Real-Time Unit Subsystem	PRUSS	—	√	√	√	√	√
Real-Time Clock	RTC	√	—	—	√	√	√
Serial ATA Controller	SATA	—	—	—	—	—	√
Serial Peripheral Interface	SPI	√	√	√	√	√	√
System Configuration Module	SCM	√	√	√	√	√	√
64-Bit Timer Plus	Timer	√	√	√	√	√	√
Universal Asynchronous Receiver/Transmitter	UART	√	√	√	√	√	√
Universal Parallel Port	uPP	—	—	—	√	—	√

Table 1. TMS320C674x DSP Peripherals Documentation (continued)

Peripheral/Module	Acronym	C67x					
		42	43	45	46	47	48
Universal Serial Bus 1.1 Controller	USB1.1	—	—	—	—	√	√
Universal Serial Bus 2.0 Controller	USB2.0	—	—	√	√	√	√
Video Port Interface	VPIF	—	—	—	√	—	√

Table 2. OMAP-L1x C6-Integra DSP+ARM Processor Peripherals Documentation

Peripheral/Module	Acronym	L132	L137	L138
ARM Interrupt Controller	AINTC	√	√	√
DDR2/mDDR Memory Controller	DDR2	√	—	√
DSP Interrupt Controller	INTC	√	√	√
Enhanced Capture Module	eCAP	√	√	√
Enhanced Direct Memory Access Controller	EDMA3	√	√	√
Enhanced High-Resolution Pulse-Width Modulator	eHRPWM	√	√	√
Enhanced Quadrature Encoder Pulse Module	eQEP	—	√	—
Ethernet Media Access Controller/ Management Data Input/Output Module	EMAC/MDIO	√	√	√
External Memory Interface A	EMIFA	√	√	√
External Memory Interface B	EMIFB	—	√	—
General-Purpose Input/Output	GPIO	√	√	√
Host Port Interface	HPI	—	√	√
Inter-Integrated Circuit Module	I2C	√	√	√
Internal Direct Memory Access Controller	IDMA	√	√	√
Liquid Crystal Display Controller	LCDC	—	√	√
Multichannel Audio Serial Port	McASP	√	√	√
Multichannel Buffered Serial Port	McBSP	√	—	√
Multimedia Card/Secure Digital Card Controller	MMC/SD	√	√	√
Phase-Locked Loop Controller	PLLC	√	√	√
Power-Down Controller	PDC	√	√	√
Power and Sleep Controller	PSC	√	√	√
Programmable Real-Time Unit Subsystem	PRUSS	√	√	√
Real-Time Clock	RTC	√	√	√
Serial ATA Controller	SATA	—	—	√
Serial Peripheral Interface	SPI	√	√	√
System Configuration Module	SCM	√	√	√
64-Bit Timer Plus	Timer	√	√	√
Universal Asynchronous Receiver/Transmitter	UART	√	√	√
Universal Parallel Port	uPP	—	—	√
Universal Serial Bus 1.1 Controller	USB1.1	—	√	√
Universal Serial Bus 2.0 Controller	USB2.0	√	√	√
Video Port Interface	VPIF	—	—	√

2 ARM Interrupt Controller (AINTC)

The ARM interrupt controller (AINTC) controls the system interrupt mapping to the host interrupt interface. System interrupts are generated by the peripheral modules. The AINTC receives the system interrupts and maps them to internal channels. The channels are used to group interrupts together and to prioritize them. These channels are then mapped onto the host interface that is typically a smaller number of host interrupts or a vector input. Interrupts from system side are active high in polarity. Also, they are pulse type of interrupts.

The AINTC encompasses many functions to process the system interrupts and prepare them for the host interface. These functions are: processing, enabling, status, channel mapping, host interrupt mapping, prioritization, vectorization, debug, and host interfacing.

3 DDR2/mDDR Memory Controller

The DDR2/mDDR memory controller is used to interface with JESD79D-2 standard compliant DDR2 SDRAM devices and standard mobile DDR (mDDR) SDRAM devices. Memories types such as DDR1 SDRAM, SDR SDRAM, SBSRAM, and asynchronous memories are not supported. The DDR2/mDDR memory is the major memory location for program and data storage.

The DDR2/mDDR memory controller supports the following features:
* JESD79D-2 standard compliant DDR2 SDRAM
* Standard compliant mobile DDR (mDDR)
* Data bus width of 16 bits
* CAS latencies:
    – DDR2: 2, 3, 4, and 5
    – mDDR: 2 and 3
* Internal banks:
    – DDR2: 1, 2, 4, and 8
    – mDDR: 1, 2, and 4
* Burst length: 8
* Burst type: sequential
* 1 CS signal
* Page sizes: 256, 512, 1024, and 2048
* SDRAM auto-initialization
* Self-refresh mode
* Partial array self-refresh (for mDDR)
* Power-down mode
* Prioritized refresh
* Programmable refresh rate and backlog counter
* Programmable timing parameters
* Little-endian mode

4   DSP Interrupt Controller (INTC)

The TMS320C674x megamodule includes a DSP interrupt controller (INTC) to manage CPU interrupts. The INTC maps DSP device events to 12 CPU interrupts. The INTC is fully described in the *TMS320C674x DSP Megamodule Reference Guide* (SPRUFK5).

5   Enhanced Capture Module (eCAP)

Uses for the enhanced capture module (eCAP) include:

*   Sample rate measurements of audio inputs
*   Speed measurements of rotating machinery (for example, toothed sprockets sensed via Hall sensors)
*   Elapsed time measurements between position sensor pulses
*   Period and duty cycle measurements of pulse train signals
*   Decoding current or voltage amplitude derived from duty cycle encoded current/voltage sensors

6   Enhanced Direct Memory Access (EDMA3) Controller

The enhanced direct memory access (EDMA3) controller's primary purpose is to service user-programmed data transfers between two memory-mapped slave endpoints on the device. Typical usage includes, but is not limited to:

*   Servicing software driven paging transfers (for example, from external memory to internal device memory
*   Servicing event driven peripherals, such as a serial port
*   Performing sorting or subframe extraction of various data structures
*   Offloading data transfers from the main device CPU(s) or DSP(s) (See your device-specific data manual for specific peripherals that are accessible via EDMA3. See the section on SCR connectivity in the device data manual for EDMA3 connectivity.)

The EDMA3 controller consists of two principal blocks:

*   The EDMA3 channel controller (EDMA3CC) serves as the user interface for the EDMA3 controller. The EDMA3CC includes parameter RAM (PaRAM), channel control registers, and interrupt control registers. The EDMA3CC serves to prioritize incoming software requests or events from peripherals, and submits transfer requests (TR) to the EDMA3 transfer controller.
*   The EDMA3 transfer controller(s) (EDMA3TC) are responsible for data movement. The transfer request packets (TRP) submitted by the EDMA3CC contains the transfer context, based on which the transfer controller issues read/write commands to the source and destination addresses programmed for a given transfer.

7   Enhanced High-Resolution Pulse-Width Modulator (eHRPWM)

The enhanced high-resolution pulse-width modulator (eHRPWM) supports the following features:

- Dedicated 16-bit time-base counter with period and frequency control.
- Two PWM outputs (EPWMxA and EPWMxB).
- Asynchronous override control of PWM signals through software.
- Programmable phase-control support for lag or lead operation relative to other ePWM modules.
- Hardware-locked (synchronized) phase relationship on a cycle-by-cycle basis.
- Dead-band generation with independent rising and falling edge delay control.
- Programmable trip zone allocation of both cycle-by-cycle trip and one-shot trip on fault conditions.
- A trip condition can force either high, low, or high-impedance state logic levels at PWM outputs.
- Programmable event prescaling minimizes CPU overhead on interrupts.
- PWM chopping by high-frequency carrier signal, useful for pulse transformer gate drives.

The enhanced high-resolution pulse-width modulator (eHRPWM) extends the time resolution capabilities of the conventionally derived digital pulse-width modulator (PWM). HRPWM is typically used when PWM resolution falls below ~9-10 bits. The key features of HRPWM are:

- Extended time resolution capability
- Used in both duty cycle and phase-shift control methods
- Finer time granularity control or edge positioning using extensions to the Compare A and Phase registers
- Implemented using the A signal path of PWM, that is, on the EPWMxA output. EPWMxB output has conventional PWM capabilities
- Self-check diagnostics software mode to check if the micro edge positioner (MEP) logic is running optimally

8   Enhanced Quadrature Encoder Pulse Module (eQEP)

The enhanced quadrature encoder pulse module (eQEP) contains the following major functional units:

- Programmable input qualification for each pin (part of the GPIO MUX)
- Quadrature decoder unit (QDU)
- Position counter and control unit for position measurement (PCCU)
- Quadrature edge-capture unit for low-speed measurement (QCAP)
- Unit time base for speed/frequency measurement (UTIME)
- Watchdog timer for detecting stalls (QWDOG)

9   Ethernet Media Access Controller (EMAC)/Management Data Input/Output (MDIO) Module

The ethernet media access controller (EMAC)/management data input/output (MDIO) module is used to move data between the device and another host connected to the same network, in compliance with the Ethernet protocol. The three main functional modules of the EMAC/MDIO peripheral are:

- EMAC control module
- EMAC module
- MDIO module

The EMAC control module is the main interface between the device core processor to the EMAC and MDIO modules. The EMAC control module controls device interrupts and incorporates an 8k-byte internal RAM to hold EMAC buffer descriptors (also known as CPPI RAM).

The MDIO module implements the 802.3 serial management interface to interrogate and control up to 32 Ethernet PHYs connected to the device by using a shared two-wire bus. Host software uses the MDIO module to configure the autonegotiation parameters of each PHY attached to the EMAC, retrieve the negotiation results, and configure required parameters in the EMAC module for correct operation. The module is designed to allow almost transparent operation of the MDIO interface, with very little maintenance from the core processor.

The EMAC module provides an efficient interface between the processor and the network. The EMAC on this device supports 10Base-T (10 Mbits/sec) and 100BaseTX (100 Mbits/sec), half-duplex and full-duplex mode, and hardware flow control and quality-of-service (QOS) support.

10   External Memory Interface A (EMIFA)

The external memory interface A (EMIFA) controller is complaint with the JESD21-C SDR SDRAM memories utilizing 16-bit data bus of EMIFA memory controller. The purpose of this EMIFA is to provide a means for the CPU to connect to a variety of external devices including:

- Single data rate (SDR) SDRAM
- Asynchronous devices including NOR Flash, NAND Flash, and SRAM

The most common use for the EMIFA is to interface with both a flash device and an SDRAM device simultaneously.

11   External Memory Interface B (EMIFB)

The external memory interface B (EMIFB) controller is complaint with the JESD21-C SDR SDRAM memories utilizing either 32-bit or 16-bit of the EMIFB memory controller data bus. The purpose of this EMIFB is to provide a means for the CPU to connect to a variety of external devices including:

- Single data rate (SDR) SDRAM/ mobile SDR SDRAM

12   General-Purpose Input/Output (GPIO)

The general-purpose input/output (GPIO) peripheral provides dedicated general-purpose pins that can be configured as either inputs or outputs. When configured as an output, you can write to an internal register to control the state driven on the output pin. When configured as an input, you can detect the state of the input by reading the state of an internal register.

The GPIO peripheral consists of the following features:

*   Output set/clear functionality through separate data set and clear registers allows multiple software processes to control GPIO signals without critical section protection.
*   Set/clear functionality through writing to a single output data register is also supported.
*   Output registers that can be read to reflect output drive status.
*   Input registers that can be read to reflect pin status.
*   All GPIO signals can be used as interrupt sources with configurable edge detection.
*   All GPIO signals can be used to generate events to the EDMA.

13   Host Port Interface (HPI)

The host port interface (HPI) provides a parallel port interface through which an external host processor can directly access the processor's resources (configuration and program/data memories). The external host device is asynchronous to the CPU clock and functions as a master to the HPI interface. The HPI enables a host device and the processor to exchange information via internal or external memory. Dedicated address (HPIA) and data (HPID) registers within the HPI provide the data path between the external host interface and the processor resources. An HPI control register (HPIC) is available to the host and the CPU for various configuration and interrupt functions. In addition to data transfer, the host can also use the HPI to bootload the processor by downloading program and data information to the processor's memory after power-up.

14   Inter-Integrated Circuit (I2C) Peripheral

The inter-integrated circuit (I2C) peripheral provides an interface between this device and other devices that are compliant with the I2C-bus specification and connected by way of an I2C-bus. External components that are attached to this two-wire serial bus can transmit and receive data that is up to eight bits wide both to and from the device through the I2C peripheral.

The I2C peripheral has the following features:

*   Compliance with the Philips Semiconductors I2C-bus specification (version 2.1)
*   2-bit to 8-bit format transfer
*   Free data format mode
*   One read DMA event and one write DMA event that the DMA can use
*   Seven interrupts that the CPU can use
*   Peripheral enable/disable capability

15   Internal Direct Memory Access (IDMA) Controller

The internal direct memory access (IDMA) controller in the TMS320C674x megamodule allows rapid data transfers between all local memories. It provides a fast way to page code and data sections into any memory-mapped RAM local to the C674x megamodule. The key advantage of the IDMA controller is that it allows for transfers between slower (level 2: L2) and faster (level 1: L1D, L1P) memory. The IDMA controller can provide lower latency than the cache controller since the transfers take place in the background of CPU operation, thereby removing stalls due to cache.

In addition, the IDMA controller facilitates rapid programming of peripheral configuration registers accessed through the external configuration space (CFG) port of the C674x megamodule. The IDMA controller view of the external configuration space that has a 32-word granularity and allows any register within a 32-word block to be individually accessed.

The IDMA is described in the *TMS320C674x DSP Megamodule Reference Guide* (SPRUFK5).

16   Liquid Crystal Display Controller (LCDC)

The liquid crystal display controller (LCDC) is capable of supporting an asynchronous (memory-mapped) LCD interface and a synchronous (raster-type) LCD interface.

The LCD controller consists of two independent controllers, the Raster Controller and the LCD Interface Display Driver (LIDD) controller. Each controller operates independently from the other and only one of them is active at any given time.

- The Raster Controller handles the synchronous LCD interface. It provides timing and data for constant graphics refresh to a passive display. It supports a wide variety of monochrome and full-color display types and sizes by use of programmable timing controls, a built-in palette, and a gray-scale/serializer. Graphics data is processed and stored in frame buffers. A frame buffer is a contiguous memory block in the system. A built-in DMA engine supplies the graphics data to the Raster engine which, in turn, outputs to the external LCD device.

- The LIDD Controller supports the asynchronous LCD interface. It provides full-timing programmability of control signals (CS, WE, OE, ALE) and output data.

17   Multichannel Audio Serial Port (McASP)

The multichannel audio serial port (McASP) functions as a general-purpose audio serial port optimized for the needs of multichannel audio applications. The McASP is useful for time-division multiplexed (TDM) stream, Inter-IC Sound (I2S) protocols, and intercomponent digital audio interface transmission (DIT).

The McASP consists of transmit and receive sections that may operate synchronized, or completely independently with separate master clocks, bit clocks, and frame syncs, and using different transmit modes with different bit-stream formats. The McASP module also includes up to 16 serializers that can be individually enabled to either transmit or receive. In addition, all of the McASP pins can be configured as general-purpose input/output (GPIO) pins.

18   Multichannel Buffered Serial Port (McBSP)

The primary use for the multichannel buffered serial port (McBSP) is for audio interface purposes. The primary audio modes that are supported are the AC97 and IIS modes. In addition to the primary audio modes, the McBSP can be programmed to support other serial formats but is not intended to be used as a high-speed interface.

The McBSP provides the following functions:

*   Full-duplex communication
*   Double-buffered data registers, which allow a continuous data stream
*   Independent framing and clocking for receive and transmit
*   Direct interface to industry-standard codecs, analog interface chips (AICs), and other serially connected analog-to-digital (A/D) and digital-to-analog (D/A) devices
*   External shift clock or an internal, programmable frequency shift clock for data transfer

In addition, the McBSP has the following capabilities:

*   Direct interface to:
    –   T1/E1 framers
    –   MVIP switching compatible and ST-BUS compliant devices
    –   IOM-2 compliant devices
    –   AC97 compliant devices (the necessary multiphase frame synchronization capability is provided)
    –   IIS compliant devices
    –   SPI™ devices
*   Multichannel transmit and receive of up to 128 channels
*   A wide selection of data sizes, including 8, 12, 16, 20, 24, and 32 bits
*   µ-Law and A-Law companding
*   8-bit data transfers with the option of LSB or MSB first
*   Programmable polarity for both frame synchronization and data clocks
*   Highly programmable internal clock and frame generation
*   Additional McBSP Buffer FIFO (BFIFO):
    –   Provides additional data buffering
    –   Provides added tolerance to variations in host/DMA controller response times
    –   May be used as a DMA event pacer
    –   Independent Read FIFO and Write FIFO
    –   256 bytes of RAM for each FIFO (read and write)
    –   Option to bypass Write FIFO and/or Read FIFO, independently

19 Multimedia Card/Secure Digital (MMC/SD) Card Controller

A number of applications use the multimedia card (MMC)/secure digital (SD) card to provide removable data storage. The MMC/SD card controller provides an interface to external MMC and SD cards. The communication between the MMC/SD card controller and MMC/SD card(s) is performed according to the MMC/SD protocol.

The MMC/SD card controller has the following features:
*   Supports interface to multimedia cards (MMC)
*   Supports interface to secure digital (SD) memory cards
*   Ability to use the MMC/SD protocol and Secure Digital Input Output (SDIO) protocol
*   Programmable frequency of the clock that controls the timing of transfers between the MMC/SD controller and memory card
*   512-bit read/write FIFO to lower system overhead
*   Signaling to support enhanced direct memory access (EDMA) transfers (slave)
*   20 MHz maximum clock to MMC (specification v4.0)
*   25 MHz maximum clock to SD (specification v1.1)

20 Phase-Locked Loop Controller (PLLC)

The phase-locked loop controller (PLLC) provides clock signals to most of the components of the device through various clock dividers.

The PLLC provides the following:
*   Glitch-free transitions when clock settings are changed
*   Domain clock alignment
*   Clock gating
*   PLL power-down

21 Power-Down Controller (PDC)

The TMS320C674x megamodule includes a power-down controller (PDC). The PDC can power-down all of the following components of the C674x megamodule and internal memories of the DSP subsystem:
*   C674x CPU
*   Program memory controller (PMC)
*   Data memory controller (DMC)
*   Unified memory controller (UMC)
*   Extended memory controller (EMC)
*   Internal Direct Memory Access controller (IDMA)
*   L1P/L1D/L2 memory

The PDC is described in the *TMS320C674x DSP Megamodule Reference Guide* (SPRUFK5).

22 Power and Sleep Controller (PSC)

The power and sleep controllers (PSC) are responsible for managing transitions of system power on/off, clock on/off, resets (device level and module level). It is used primarily to provide granular power control for on chip modules (peripherals and CPU). A PSC module consists of a Global PSC (GPSC) and a set of Local PSCs (LPSCs). The GPSC contains memory mapped registers, PSC interrupts, a state machine for each peripheral/module it controls. An LPSC is associated with every module that is controlled by the PSC and provides clock and reset control. Many of the operations of the PSC are transparent to user (software), such as power on and reset control. However, the PSC module(s) also provide you with interface to control several important power, clock and reset operations.

The PSC includes the following features:
- Manages chip power-on/off
- Provides a software interface to:
  - Control module clock enable/disable
  - Control module reset
  - Control CPU local reset
- Manages on-chip RAM sleep modes (for DSP memories and L3 RAM)
- Supports IcePick emulation features: power, clock and reset

23 Programmable Real-Time Unit Subsystem (PRUSS)

The Programmable Real-Time Unit Subsystem (PRUSS) consists of:
- Two programmable real-time units (PRU0 and PRU1) and their associated memories.
- An interrupt controller (INTC) for handling system interrupt events. The INTC also supports posting events back to the device level host CPU.
- A Switched Central Resource (SCR) for connecting the various internal and external masters to the resources inside the PRUSS.

The two PRUs can operate completely independently or in coordination with each other. The PRUs can also work in coordination with the device level host CPU. This is determined by the nature of the program which is loaded into the PRUs instruction memory. Several different signaling mechanisms are available between the two PRUs and the device level host CPU.

The PRUs are optimized for performing embedded tasks that require manipulation of packed memory-mapped data structures, handling of system events that have tight realtime constraints and interfacing with systems external to the device.

The PRUSS documentation (peripheral guide) is on the external wiki at:
http://processors.wiki.ti.com/index.php/Programmable_Realtime_Unit.

24 Real-Time Clock (RTC)

The real-time clock (RTC) provides a time reference to an application running on the device. The current date and time is tracked in a set of counter registers that update once per second. The time can be represented in 12-hour or 24-hour mode. The calendar and time registers are buffered during reads and writes so that updates do not interfere with the accuracy of the time and date.

Alarms are available to interrupt the CPU at a particular time, or at periodic time intervals, such as once per minute or once per day. In addition, the RTC can interrupt the CPU every time the calendar and time registers are updated, or at programmable periodic intervals.

The real-time clock (RTC) provides the following features:

*   100-year calendar (xx00 to xx99)
*   Counts seconds, minutes, hours, day of the week, date, month, and year with leap year compensation
*   Binary-coded-decimal (BCD) representation of time, calendar, and alarm
*   12-hour clock mode (with AM and PM) or 24-hour clock mode
*   Alarm interrupt
*   Periodic interrupt
*   Single interrupt to the CPU
*   Supports external 32.768-kHz crystal or external clock source of the same frequency
*   Separate isolated power supply

25 Serial ATA (SATA) Controller

The serial ATA (SATA) controller is the successor of the parallel ATA/ATAPI controller and is used to interface to data storage devices at both 1.5 Gbits/second and 3.0 Gbits/second line speeds. This device has a built-in SATA controller with a single HBA port operating in Advanced Host Controller Interface (AHCI) mode.

The main features of the SATA controller are:

*   Synopsis DWH Serial ATA 1.5 Gbps and 3 Gbps speeds core
*   Support for the AHCI controller spec 1.1
*   Integrated TI SERDES PHY
*   Integrated Rx and Tx data buffers
*   Supports all SATA power management features
*   Internal DMA engine per port
*   Hardware-assisted native command queuing (NCQ) for up to 32 entries
*   32-bit addressing
*   Supports port multiplier with command-based switching
*   Activity LED support
*   Mechanical presence switch
*   Cold presence detect

26   Serial Peripheral Interface (SPI)

The serial peripheral interface (SPI) is a high-speed synchronous serial input/output port that allows a serial bit stream of programmed length (2 to 16 bits) to be shifted into and out of the device at a programmed bit-transfer rate. The SPI is normally used for communication between the device and external peripherals. Typical applications include interface to external I/O or peripheral expansion via devices such as shift registers, display drivers, SPI EPROMS and analog-to-digital converters.

The SPI has the following features:
*   16-bit shift register
*   16-bit Receive buffer register and 16-bit Receive buffer emulation 'alias' register
*   16-bit Transmit data register and 16-bit Transmit data and format selection register
*   8-bit baud clock generator
*   Programmable SPI clock frequency range
*   Programmable character length (2 to 16 bits)
*   Programmable clock phase (delay or no delay)
*   Programmable clock polarity (high or low)
*   Interrupt capability
*   DMA support (read/write synchronization events)

27   System Configuration Module (SCM)

The system configuration (SYSCFG) module is a system-level module containing status and top level control logic required by the device. The system configuration module consists of a set of memory-mapped status and control registers, accessible by the CPU, supporting all of the following system features, and miscellaneous functions and operations:
*   Device Identification
*   Device Configuration
*   Master Priority Control
*   Emulation Control
*   Special Peripheral Status and Control
    –   Locking of PLL control settings
    –   Default burst size configuration for EDMA3 transfer controllers
    –   Event source selection for the eCAP peripheral input capture
    –   McASP AMUTEIN selection and clearing of AMUTE
    –   USB PHY Control
    –   Clock source selection for EMIFA and EMIFB
    –   HPI Control
*   ARM-DSP Integration
    –   On-chip inter-processor interrupts and status for signaling between ARM and DSP

The system configuration module controls several global operations of the device; therefore, the module supports protection against erroneous and illegal accesses to the registers in its memory-map. The protection mechanisms that are present in the module are:
*   A special key sequence that needs to be written into a set of registers in the system configuration module, to allow write ability to the rest of registers in the system configuration module.
*   Several registers in the module are only accessible when the CPU requesting read/write access is in privileged mode.

28 64-Bit Timer Plus

The 64-bit Timer Plus supports four basic modes of operation: a 64-bit general-purpose (GP) timer, dual unchained 32-bit GP timers, dual chained 32-bit timers, or a watchdog timer. The GP timer modes can be used to generate periodic interrupts and DMA synchronization events. The watchdog timer mode is used to provide a recovery mechanism for the device in the event of a fault condition (such as a non-exiting code loop).

The 64-bit Timer Plus supports the following additional features over the other timers:

- External clock/event input
- Period reload
- External event capture mode
- Timer counter register read reset mode
- Timer counter capture registers
- Register for interrupt/DMA generation control and status

29 Universal Asynchronous Receiver/Transmitter (UART)

The universal asynchronous receiver/transmitter (UART) peripheral is based on the industry standard TL16C550 asynchronous communications element, which in turn is a functional upgrade of the TL16C450. Functionally similar to the TL16C450 on power up (single character or TL16C450 mode), the UART can be placed in an alternate FIFO (TL16C550) mode. This relieves the CPU of excessive software overhead by buffering received and transmitted characters. The receiver and transmitter FIFOs store up to 16 bytes including three additional bits of error status per byte for the receiver FIFO.

The UART performs serial-to-parallel conversions on data received from a peripheral device and parallel-to-serial conversion on data received from the CPU. The CPU can read the UART status at any time. The UART includes control capability and a processor interrupt system that can be tailored to minimize software management of the communications link.

30 Universal Parallel Port (uPP)

The universal parallel port (uPP) peripheral is a multichannel, high-speed parallel interface with dedicated data lines and minimal control signals. It is designed to interface cleanly with high-speed analog-to-digital converters (ADCs) or digital-to-analog converters (DACs) with up to 16-bit data width (per channel). It may also be interconnected with field-programmable gate arrays (FPGAs) or other uPP devices to achieve high-speed digital data transfer. It can operate in receive mode, transmit mode, or duplex mode, in which its individual channels operate in opposite directions.

The uPP peripheral includes an internal DMA controller to maximize throughput and minimize CPU overhead during high-speed data transmission. All uPP transactions use the internal DMA to feed data to or retrieve data from the I/O channels. The DMA controller includes two DMA channels, which typically service separate I/O channels. The uPP peripheral also supports data interleave mode, in which all DMA resources service a single I/O channel. In this mode, only one I/O channel may be used.

31   Universal Serial Bus 1.1 (USB1.1) Controller

The USB OHCI host controller (HC) is a single port controller that communicates with USB devices at the USB low-speed (1.5M bit-per-second maximum) and full-speed (12M bit-per-second maximum) data rates. It is compatible with the *Universal Serial Bus Specification Revision 2.0* and the *Open HCI—Open Host Controller Interface Specification for USB, Release 1.0a*.

The USB host controller implements the register set and makes use of the memory data structures defined in the *OHCI Specification for USB*. These registers and data structures are the mechanisms by which a USB host controller driver software package can control the USB host controller. The *OHCI Specification for USB* also defines how the USB host controller implementation must interact with those registers and data structures in system memory.

To reduce processor software and interrupt overhead, the USB host controller generates USB traffic based on data structures and data buffers stored in system memory. The USB host controller accesses these data structures without direct intervention by the processor using its bus master port. These data structures and data buffers can be located in internal or external system RAM.

32   Universal Serial Bus 2.0 (USB2.0) Controller

The universal serial bus (USB) controller complies with the USB 2.0 standard high-speed and full-speed functions and low-speed, full-speed, and high-speed limited host mode operations. It also includes support for the Session Request and Host Negotiation Protocols used in point-to-point communications, details of which are given in the USB On-The-Go supplement to the USB 2.0 specification. In addition, the four test modes for high-speed operation described in the USB 2.0 specification are supported. It also allows options that allow the USB controller to be forced into full-speed mode, high-speed mode, or host mode that may be used for debug purposes.

The USB controller provides a low-cost connectivity solution for consumer portable devices by providing a mechanism for data transfer between USB devices up to 480 Mbps. Its support for a dual-role feature allows for additional versatility supporting operation capability as a host or peripheral.

33   Video Port Interface (VPIF)

The video port interface (VPIF) is a receiver and transmitter of video data. The VPIF has two input channels that receive video byte stream data and two output channels that video byte stream data is asserted. Channels 0 and 1 have the same architecture, and channels 2 and 3 have the same architecture, this allows you to invert the task of each channel (channels 0 and 1 for input, and channels 2 and 3 for output).

The VPIF is designed to support the following features (note that some device designs may support reduced features because of system-level performance limitations):

*   ITU-BT.656 format
*   ITU-BT.1120 and SMTPE 296 formats
*   Raw data capture
*   VBI data storage
*   Clipping of output data (to eliminate FFh and 00h values)